收稿日期：00 Month0000 修订日期：00 Month 0000 接受日期：00 Month0000DOI：xxx/xxxx

期刊

大规模偏差 加速神经网络训练的方法

Sreelekha Guggilam*[2, 1] | Varun Chandola[2, 3] | Abani Patra[4]

[1]国家安全与科学局，橡树岭国家实验室，美国马萨诸塞州[2]计算数据科学与工程，纽约州立大学布法罗分校，美国纽约州[3]计算机科学与工程，纽约州立大学布法罗分校（纽约州立大学），美国纽约州[4]数据密集型研究中心，塔夫茨大学，美国马萨诸塞州

通信作者
电子邮箱：{guggilams@ornl.gov, chandola@buffalo.edu, abani.patra@tufts.edu}

arXiv:2303.00954v1 [cs.LG] 2 Mar 2023

摘要

人工神经网络（ANN）需要大量数据进行训练。然而，在分类模型中，大多数数据特征通常相似，这可能会增加训练时间，却无法显著提升性能。因此，我们假设，使用更具代表性的样本训练人工神经网络，或许能找到更高效的方法。为此，我们提出 LAD 改进迭代训练（LIIT），这是一种创新的人工神经网络训练方法，利用大偏差原理，在快速高效的设置下生成并迭代更新训练样本。这是一项探索性研究，未来还有大量拓展机会。本论文介绍了这项正在进行的研究，并总结了以下贡献：(1) 我们提出了一种基于大偏差理论的创新人工神经网络训练方法 LIIT，无需额外降维即可研究高维数据。(2) LIIT 方法采用改进训练样本（MTS），并通过基于 LAD 异常分数的采样策略生成和迭代更新该样本。(3) MTS 样本旨在充分代表训练数据，其中包含每个类别中最异常的观测值。这确保了模型能够用更少的样本学习不同模式和特征。(4) 我们研究了 LIIT 训练的人工神经网络与传统批量训练的人工神经网络的分类性能。

关键词：
大偏差、异常检测、高维数据、多变量时间序列

1 引言

人工神经网络（ANN）是不依赖假设的模型，能够从提供的训练数据中收集信息。凭借其设计特点，它们非常适合研究输入层和输出层之间复杂的函数依赖关系。与使用均值、协方差矩阵、概率和置信区间等指标的传统统计模型不同，人工神经网络依赖训练数据中的模式进行模型开发和拟合。尽管这有助于构建比统计模型更贴合数据的架构，尤其是在结构复杂的数据集中，但错误或不足的训练数据会造成深远影响。

本文提出了一种对训练数据进行统计增强采样的方法，并将其与一种新颖的训练方法相结合，从而确保神经网络能够更快地完成训练。本研究的贡献如下：

1. 我们提出了 LAD改进迭代训练（LIIT）策略，该策略使用改进训练样本（MTS）来训练神经网络。

2. 我们提出了四种基于 LAD 分数的采样策略，用于设计 MTS。基于大偏差原理获取 LAD 分数的计算成本很低。因此，无需额外的降维步骤，即可分析大型高维数据集，从而实现更准确且更具成本效益的评分方案。

3. 使用 MTS（一种规模较小的训练样本）可显著降低大型数据集的计算时间成本。

4. 我们针对公开可用的分类基准数据集开展了实证研究，以分析所提出方法的性能。

本文的工作仅涉及基于简单分类的神经网络。未来的工作将把这一方法扩展到更复杂的人工神经网络。

2 相关工作

在本节中，我们　简要概述对训练样本的敏感性以及神经网络训练的速度　.

人工神经网络在通用分类方面功能强大。然而，其出色的性能通常在很大程度上依赖于庞大的训练集。大量研究探讨了训练数据规模对神经网络学习的影响[6, 2]。特别是，显而易见的是，较小的训练数据规模会导致模型效率降低。然而，在大规模数据集上进行训练所产生的巨大计算开销，使得改进训练实践变得至关重要，尤其是对于在线或实时模型而言。

目前已有许多旨在构建更快神经网络的方法。例如，Wang 等人 [7] 在深度神经网络中使用批量归一化，以提高收敛速度。Zhong 等人 [8] 采用敏捷卷积神经网络 SatCNN 进行图像分类，通过较小的卷积核和较深的卷积层实现快速且有效的学习。然而，这些工作仅限于特定领域的问题，难以轻松扩展到其他数据类型。

另一种提高训练速度的方法是修改训练样本。例如，Shanker 等人的研究 [5] 探讨了数据标准化对神经网络学习的影响。Kavzoglu [3] 强调训练样本的特征，并采用代表性训练来提高分类效果。然而，这些方法未能研究较小数据量对模型性能和效率的影响。

在论文这一部分，我们提出了一种可推广到不同领域的新型训练策略。该方法旨在用较小的样本复现训练特征的真实表示，从而加快训练和收敛。由于能够恰当地表示极端观测值，该方法可确保更快的学习速度，同时保持有竞争力的性能。

3 方法论

分类模型最重要的方面是每个类别的代表性训练样本是否充分。尽管训练数据的规模相当重要，但在涉及大量类别时，获取大量具有代表性的训练数据可能并不现实。具体而言，由于每个真实类别中的大多数观测值都具有相似的特征，增加多个样本所带来的新颖信息/模式价值较低。本节将简要介绍传统的批量训练方法，随后介绍 LAD改进迭代训练方法。我们将介绍 LIIT训练中使用的 4 种采样策略及其相应算法。

3.1 定义与术语

在介绍具体方法之前，我们先列出本研究中使用的术语及其相应定义。

定义 1. LAD 分数 是大偏差异常检测（LAD）为数据中的每个观测值生成的异常分数。

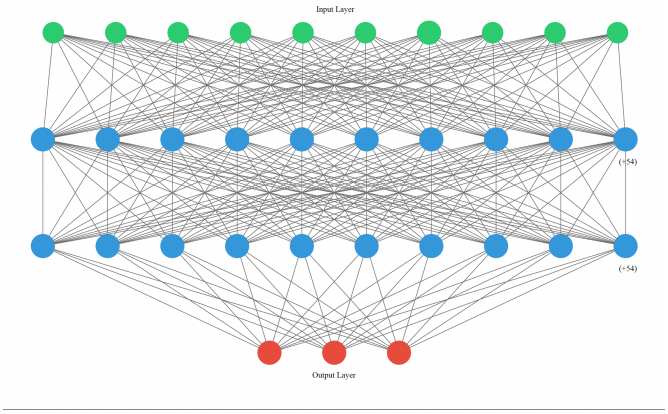


图 1 简单分类神经网络：图中展示了一个用于将数据分类为 3 个类别的全连接神经网络。该网络接收一个维度为 10 的输入，并返回用于分配各类别的分数。

定义 2. 完整训练数据 是人工神经网络（ANN）可用的完整训练数据集。需要注意的是，在 LIIT 方法中，可能只使用完整训练数据的一个子集来训练人工神经网络（ANN）。因此，我们使用不同的术语，以便将其与训练数据区分开来。

定义 3　　. 批量训练 是一种传统的人工神经网络训练方法，使用训练数据的小批量进行训练　　.

定义 4. 修改后的训练样本（MTS）是使用特定采样算法从训练数据中生成的较小样本。

定义 5. LAD 改进　　改进迭代训练（LIIT）是用于训练人工神经网络的一种新颖的改进批量训练方法　　.

3.2 分类神经网络

在此分析中，我们研究一种基本的分类算法。图 1 展示了一个简单的三层全连接神经网络的架构。

该模型使用全部训练样本进行训练，收敛准则设为验证损失为零，训练轮次为 5，最大训练轮次设为 180。三个连续的全连接层分别使用三种不同的激活函数：ReLU、双曲正切和 Softmax。选择一个简单模型，是为了验证论文这一部分所介绍的代表性采样策略的概念。还需要进一步研究，以了解模型选择与训练采样技术之间的关系。

3.3 神经网络的LAD改进迭代训练

传统上，在批量训练中，完整训练数据会被划分为较小的样本或批次。人工神经网络依次从每个批次中学习，直到完整训练数据中的所有观测值都处理完毕，如图2所示。在 LIIT训练中，我们从完整训练数据中迭代设计并更新修正训练样本 MTS。在每次迭代中，我们使用批量训练在 MTS 上训练人工神经网络，直到收敛。随后，使用完整训练数据对该部分训练模型进行测试，以识别潜在的学习缺陷。由于当前工作仅限于分类模型，学习缺陷包括误分类数据。然后利用误分类数据推导出更新后的 MTS，并使用该 MTS 重新训练人工神经网络。该过程如图3所示。这一方法受到提升技术的启发 [4]，其中子集的创建取决于先前的模型。然而，与提升法的设置不同，我们重新训练的是同一个人工神经网络。 [1]

[1]LIIT 方法与神经网络每个轮次内的批量训练非常相似。

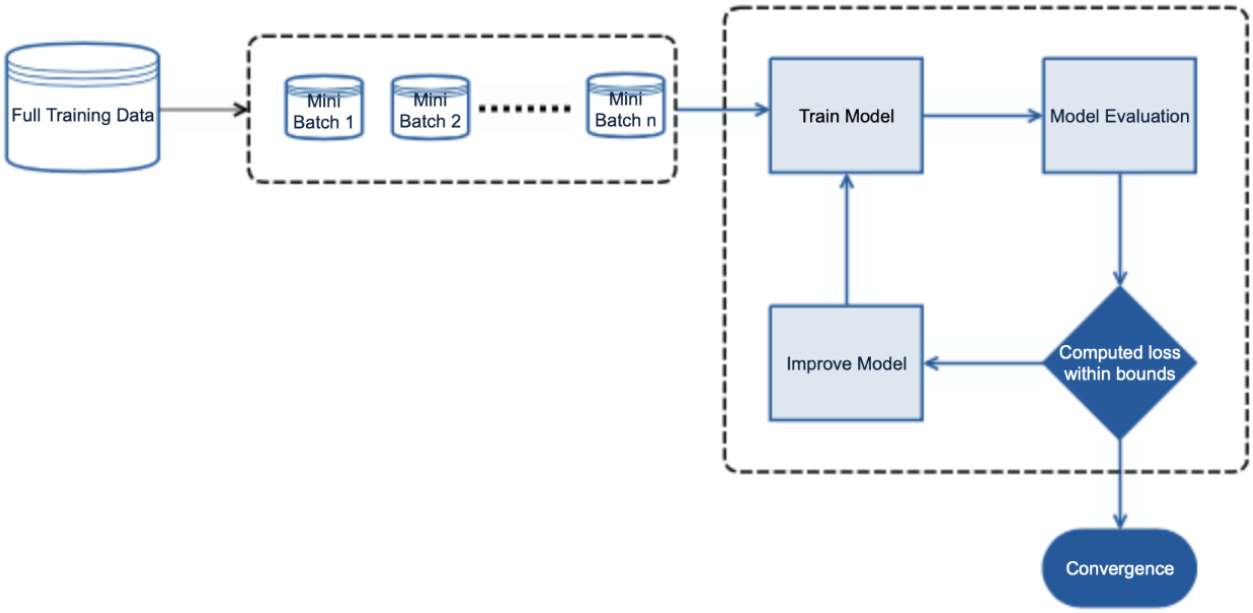


图2 小批量训练算法

为了确定并提取 MTS 样本，可以使用任何采样算法。然而，为了确保良好的代表性，我们设计了四种基于 LAD 分数的采样算法，以及作为基线使用的随机采样方法。以下是我们在分析中使用的采样策略：

1. **仅 LAD 异常（重复条目）**：将每个真实类别中异常分数最高的观测值加入训练批次。如果模型经过多次重新训练后仍无法对其进行分类，那么在后续迭代中可以多次添加同一观测值。参见算法 1。

2. **LAD 异常 + 正常（唯一条目）**：针对每个真实类别，分别抽取异常分数较高和较低的观测值，数量相等。最终训练批次包含一组唯一观测值，不含重复条目。参见算法 2。

3. **仅 LAD 异常（唯一条目）**：此方法与**仅 LAD 异常（重复条目）**类似。将每个真实类别中异常分数最高的观测值加入训练批次。不过，最终训练批次包含一组唯一观测值，不含重复条目。参见算法 3。

4. **LAD 分位数样本（重复条目）**：针对每个真实类别，使用异常分数的不同分位数抽取观测值。训练批次中会保留多个副本，以确保每个已知真实类别中代表性不足的潜在类别得到相应权重。参见算法 4。

5。**随机**：在此模型中，我们从可用数据中进行随机采样。参见算法 5.

在这部分中，我们在每次迭代时从完整训练数据中采样 $\sim 5-6\%$，随后将其添加到修改后的训练样本中。为便于分析，我们确保所有真实类别的权重相等。LIIT 方法通过 6 次迭代（1 次初始迭代和 5 次更新）实现，最终使 LIIT 方法使用的完整训练样本达到 $\sim 30\%$。

4 实验

本节评估使用 LAD 子采样数据训练的简单神经网络在真实数据上的分类性能。我们重点关注不同训练和采样设置下神经网络的性能。

我们进行了以下实验来研究该模型：

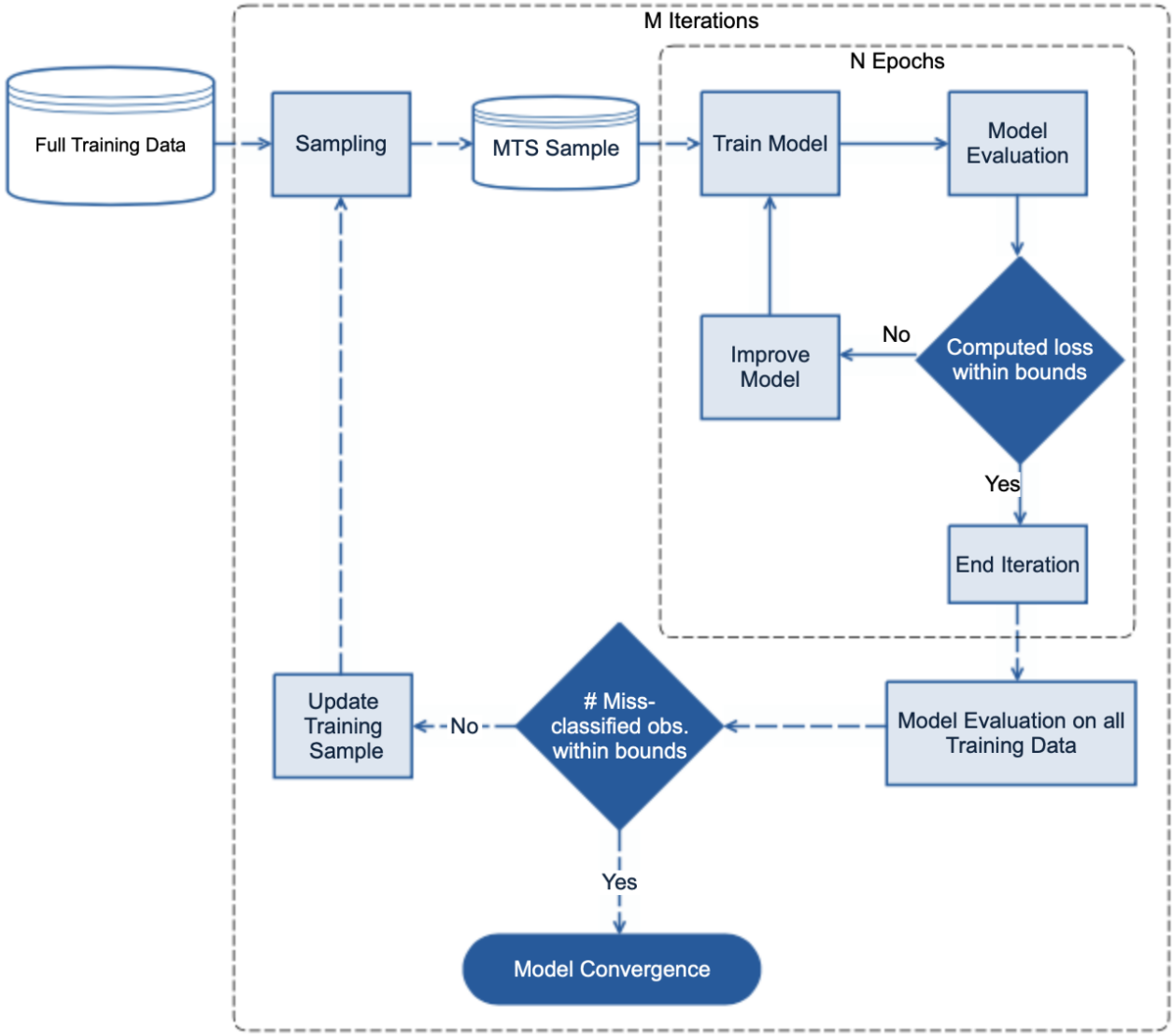


图3 LIIT训练算法

1. 计算开销：将 LIIT 训练的人工神经网络模型在较小训练样本集上训练并更快收敛的能力，与完全训练模型进行比较。2. 分类性能：研究子采样模型在多个基准数据集上的整体性能。在这项分析中，我们采用曲线下面积（AUC）作为分类性能指标。3. 对扰动的稳定性：向测试数据中加入最高达 8% 的扰动，以研究所有模型性能的变化。

为确保比较公平，使用完整训练数据训练的人工神经网络模型（即完整模型）的训练轮次上限固定为 180；所有采用 LIIT 训练的人工神经网络，每次迭代训练 30 轮（完整训练共计 180 轮）。对于每个训练好的人工神经网络，我们进行了 5 次独立的重复运行以评估其性能。所有评估结果均报告平均值。

算法 1 仅 LAD 异常检测（重复条目）

输入：数据集 X，大小为 (n, d)，迭代次数 N_{iter}，阈值 th，数据中的真实类别数 K，每个类别的样本量 c_{size}，迭代次数 i_{iter}，人工神经网络分类模型 $model_{liit}$。

初始化：
将数据拆分为 x_{train}，x_{test}，x_{val}，y_{train}，y_{test}，y_{val}（训练集、测试集和验证集）
计算 LAD 分数 ana_{score} 对于训练数据中的所有观测值，即
$ana_{score} = LAD(x_{train}, y_{train})$

1: $MTS = []$（创建空的 MTS 样本索引列表）
2: **对**每个类别 k**执行**
3: 生成类别中所有观测值的索引列表 k , ind_k
4: 每个类别的子集异常分数

$$ana_{score_k} = ana_{score}[ind_k]$$

5: 找出异常分数最低的前 c_{size} 个观测值，并将它们添加到样本 MTS 中，即（最不异常的观测）
6: **对**每次迭代 $i \leq i_{iter}$ **do**
7: 使用批量训练在 MTS 上训练人工神经网络，

$$model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$$

8: 在 x_{train}, $z_{pred} = model_{liit}.predict(x_{train})$上预测模型分类
9: 找出训练数据中所有误分类观测值的索引

$$err_{inds} = np.where(z_{pred} != y_{train})$$

10: **对**每个类别 k**执行**
11: 识别所有误分类观测 ind_{err_k}
12: 类别 k中误分类数据的子集异常分数

$$ana_{err_k} = ana_{score}[ind_{err_k}]$$

13: 从 ind_{err_k} 中识别具有最高异常分数的 c_{size} 观测值，即（最异常观测），并将它们添加到 MTS 样本中。
14: **结束循环**
15: **结束循环**
16: **结束循环**

4.1 数据集

我们选取了 UCI-ML 仓库中的多种公开基准数据集 [1]）（见表 1），用于实验评估。训练、测试和验证数据分别随机划分为数据总量的 80%、10% 和 10%。

4.1.1 计算时间

在本节中，我们考察每个人工神经网络在数据集上的训练耗时。由于 LIIT 训练的人工神经网络仅使用完整训练数据的三分之一，因此其训练时间显然低于完整模型的训练时间。图 4 清楚地展示了这一点。

4.1.2 模型性能

鉴于经过 LIIT 训练的人工神经网络模型在计算方面明显优于完整模型，我们接下来考察其在大量基准分类数据集上的总体分类性能。表 2 展示了这些模型在每个数据集上的性能。我们使用曲线下面积（AUC）作为评估指标来研究分类性能

算法 2 LAD 异常 +正常（唯一条目）

输入：数据集 X，大小为 (n,d)，迭代次数为 N_{iter}，阈值为 th，数据中的真实类别数为 K，每个类别的样本量为 c_{size}，迭代次数为 i_{iter}，人工神经网络分类模型 $model_{liit}$。

初始化：

将数据拆分为 x_{train}、x_{test}、x_{val}、y_{train}、y_{test}、y_{val}（训练、测试和验证）

推导 LAD 分数 ana_{score} 对于训练数据中的所有观测值，即

$ana_{score} = LAD(x_{train}, y_{train})$

$MTS = []$ (创建空的 MTS 样本索引列表)

对每个类别 k**执行**

生成类别中所有观测值的索引列表 k，ind_k

每个类别的子集异常分数

$$ana_{score_k} = ana_{score}[ind_k]$$

找出异常分数最低的前 c_{size} 个观测值，并将它们添加到 MTS 样本，即（最不异常的观测）

对于每次迭代 $i \le i_{iter}$ **do**

在 MTS 上使用批量训练来拟合人工神经网络，

$$model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$$

在 x_{train}，$z_{pred} = model_{liit}.predict(x_{train})$上预测模型分类

找出训练数据中所有误分类观测值的索引

$$err_{inds} = np.where(z_{pred} != y_{train})$$

对于每个类别 k**执行**

识别所有误分类观测值 ind_{err_k}

类别 k中误分类数据的子集异常分数

$$ana_{err_k} = ana_{score}[ind_{err_k}]$$

识别 c_{size}从 ind中找出异常分数最低和最高的各/2 个观测值 err_k 即（最异常的以及最不异常的观测值），并将它们添加到 MTS 样本索引中。

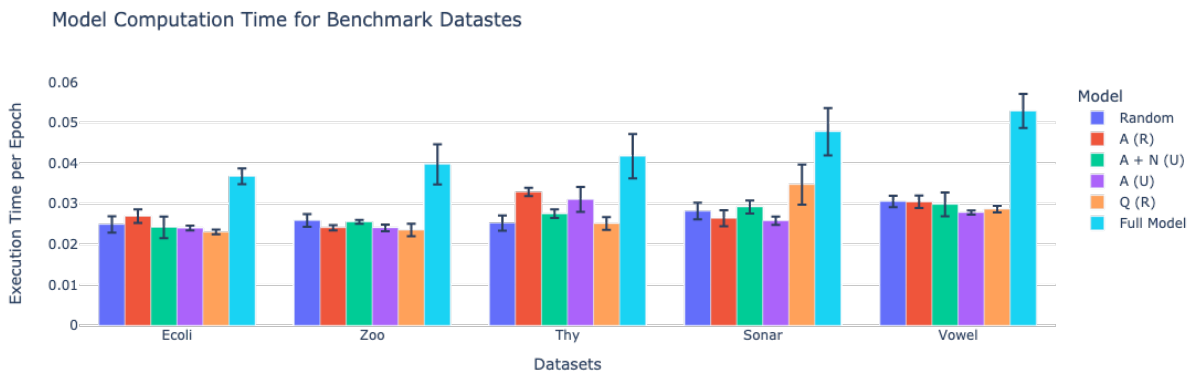
结束循环

移除更新后的修改后的训练样本中的重复索引，

$$MTS = unique(MTS)$$

结束循环

结束循环



(a)

算法 3 仅 LAD 异常检测（唯一条目）

输入：数据集 X，大小为 (n,d)，迭代次数 N_{iter}，阈值 th，数据中的真实类别数 K，每个类别的样本量各类别 c_{size}，迭代次数 i_{iter}，人工神经网络分类模型 $model_{liit}$.

初始化：

将数据拆分为 x_{train}、x_{test}、x_{val}、y_{train}、y_{test}、y_{val}（训练、测试和验证）

计算 LAD 分数 ana_{score} 针对训练数据中的所有观测值，即

$ana_{score} = LAD(x_{train}, y_{train})$

$MTS = []$（创建空的 MTS 样本索引列表）

对每个类别 k**执行**

生成类别中所有观测值的索引列表 k，ind_k

每个类别的子集异常分数

$$ana_{score_k} = ana_{score}[ind_k]$$

找出异常分数最低的前 c_{size} 个观测值，并将它们添加到样本 MTS 中，即（最不异常的观测）

对每次迭代 $i \leq i_{iter}$ **do**

使用批量训练，在 MTS 上训练人工神经网络，

$$model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$$

在 x_{train}, $z_{pred} = model_{liit}.predict(x_{train})$上预测模型分类

找出训练数据中所有误分类观测值的索引

$$err_{inds} = np.where(z_{pred} != y_{train})$$

对每个类别 k**执行**

识别所有误分类观测 ind_{err_k}

类别中的误分类数据的子集异常分数 k

$$ana_{err_k} = ana_{score}[ind_{err_k}]$$

从 ind_{err_k} 中识别异常分数最高的 c_{size} 个观测值，即（最异常观测），并将其添加到该样本中。

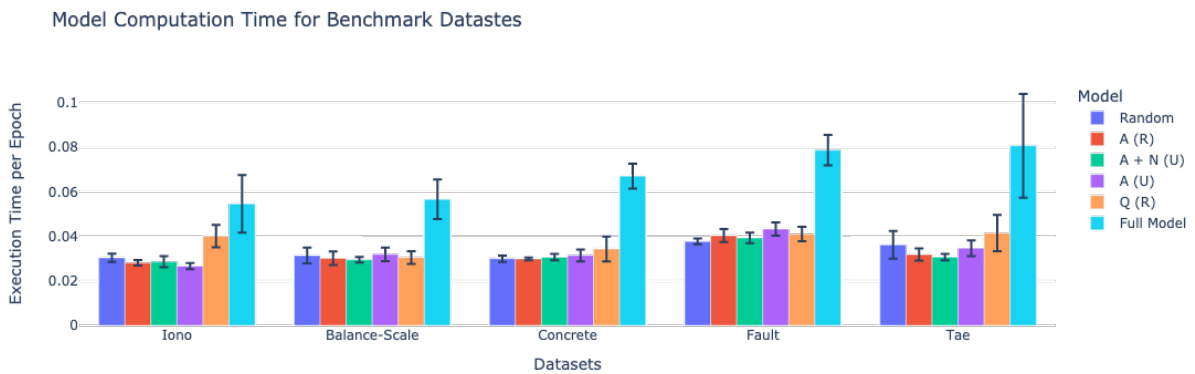
结束循环

从更新后的修改后的训练样本中移除重复索引，

$$MTS = unique(MTS)$$

结束循环

结束循环



(b)

算法 4 LAD 分位数样本（重复条目）

输入：大小为 X 的数据集，迭代次数 N_{iter}，阈值 th，数据中的真实类别数 K，每个类别的样本量每个类别 c_{size}，迭代次数 i $_{iter}$，人工神经网络分类模型 $model$ $_{liit}$.

初始化：
将数据拆分为 x_{train}，x_{test}，x_{val}，y_{train}，y_{test}，y_{val}（训练、测试和验证）
计算 LAD 分数 ana $_{score}$ 对于训练数据中的所有观测值，即
$ana_{score} = LAD(x_{train}, y_{train})$

1: $MTS = []$（创建空的 MTS 样本索引列表）
2: **对**每个类别 k**执行**
3: 生成类别中所有观测值的索引列表 k $, ind_k$
4: 每个类别的子集异常分数

$$ana_{score_k} = ana_{score}[ind_k]$$

5: 找出异常分数最低的前 c_{size} 个观测值，并将它们添加到样本 MTS 中，即（最不异常的观测）
6: **对于**每次迭代 $i \leq i$ $_{iter}$ **do**
7: 在 MTS 上使用批量训练来拟合人工神经网络，

$$model_{liit}.fit(x_{train}[MTS], \ y_{train}[MTS])$$

8: 在 x_{train}，$z_{pred} = model_{liit}.predict(x_{train})$上预测模型分类
9: 找出训练数据中所有误分类观测值的索引

$$err_{inds} = np.where(z_{pred} != y_{train})$$

10: **对于**每个类别 k**执行**
11: 识别所有误分类观测 ind $_{err_k}$
12: 类别中误分类数据的子集异常分数 k

$$ana_{err_k} = ana_{score}[ind_{err_k}]$$

13: 识别与 ana_{err_k} 分数中的 c_{size} 分位数对应的观测值，并将其添加到 MTS 样本索引中。
14: **结束循环**
15: **结束循环**
16: **结束循环**

在这些模型中，可以看出，采用分位数采样和 LIIT 训练的人工神经网络模型，其表现与完全训练模型不相上下。

4.1.3 对扰动的稳定性

由于训练样本会显著影响模型的学习和性能，我们尝试考察模型在测试数据受到各种扰动时的稳定性。为此，从符合训练数据均值和方差的多元正态分布中采样随机噪声，并将其添加到测试数据中的所有观测值上。在这些设置下，我们针对所有基准数据集评估了每个人工神经网络的性能。最终的分类性能见图 5。值得注意的是，不同数据集在采用不同采样策略时，表现更好且相对更稳定。

现在，为了了解扰动对性能造成的具体变化，我们考察所有模型因添加扰动而产生的 AUC 值原始变化。图 6 展示了不同数据集的性能变化。具体而言，图 6a 和图 6b 展示了一组采用分位数（重复）采样时表现更好的数据集；图 6c 至图 6e 则展示了分别采用异常（唯一）、异常 +正态（重复）和异常（重复）采样方法时表现更好的数据集。

可以看出，在大多数数据集中，分位数样本训练模型的 AUC 均值更高，且 AUC 的偏差小于完整训练模型。

算法 5 仅 LAD 异常检测（重复条目）

输入：数据集 X，大小为 (n, d)，迭代次数 N_{iter}，阈值 th，数据中的真实类别数量 K，每个类别的样本量为每个类别 c_{size}，迭代次数 i $_{iter}$，人工神经网络分类模型 $model$ $_{liit}$.

初始化：
将数据拆分为 x_{train}，x_{test}，x_{val}，y_{train}，y_{test}，y_{val}（训练、测试和验证）
推导 LAD 分数 ana $_{score}$ 针对训练数据中的所有观测值，即
$ana_{score} = LAD(x_{train}, y_{train})$

1: $MTS = []$ (创建空的 MTS 样本索引列表)
2: **对**每个类别 k**执行**
3: 生成类别 k中所有观测值的索引列表 , ind_k
4: 每个类别的子集异常分数

$$ana_{score_k} = ana_{score}[ind_k]$$

5: 随机抽取 c_{size} 个观测值的索引，并将其样本添加到 MTS 样本中
6: **对**每次迭代 $i \leq i$ $_{iter}$ **do**
7: 使用批量训练在 MTS 上拟合人工神经网络，

$$model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$$

8: 在 x_{train}, $z_{pred} = model_{liit}.predict(x_{train})$上预测模型分类
9: 识别训练数据中所有误分类观测的索引

$$err_{inds} = np.where(z_{pred} != y_{train})$$

10: **对于**每个类别 k**执行**
11: 识别所有误分类观测 ind $_{err_k}$
12: 类别 k中误分类数据的子集异常分数

$$ana_{err_k} = ana_{score}[ind_{err_k}]$$

13: 随机抽取 c_{size} 个观测值的索引，并将其添加到 MTS 样本中
14: **结束循环**
15: **结束循环**
16: **结束循环**

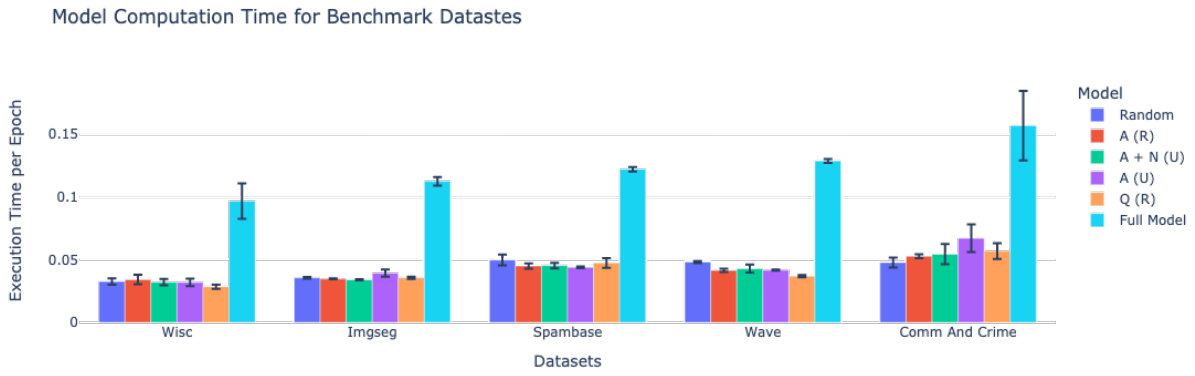
在这里可以看到，不同的 LIIT模型在不同数据集上的表现各有优劣。我们假设，数据分布和异质性对整体性能和稳定性起着重要作用。未来研究中，我们打算继续探讨这一假设。

5 结论

我们提出了一种新的训练策略，旨在提高神经网络的学习速度，同时保持模型性能。我们提出了 LAD改进迭代训练（LIIT），这是对传统批量训练方法的改进迭代版本。LIIT 方法使用基于LAD评分的采样方法生成并更新修改后的训练样本（MTS），以确保极端和罕见行为得到充分代表。具体而言，基于LAD评分的分位数采样方法使样本数据具有充分的异质性。我们在真实的基准数据集上，研究了采用 LIIT 训练的人工神经网络与使用完整训练数据训练的人工神经网络的分类性能。尽管目前的研究仅涉及简单分类神经网络，但这项工作具有巨大的研究潜力。LIIT 训练方法结合特定的 LAD采样方法，或许能够根据数据特征充分发挥数据集的性能。未来的研究或许有助于理解数据异质性和采样方法对人工神经网络性能的影响。

Name	*N*	*d*	*c*
大肠杆菌	336	7	8
图像分割	2310	18	7
Skin	245057	4	2
航天飞机	58000	10	2
Wisc	699	9	2
Iono	351	33	2
Zoo	101	16	7
字母	20000	16	26
通信与犯罪	1994	102	2
元音	990	10	11
故障	1941	28	2
声呐	208	60	2
平衡秤	625	4	3
Pageb	5473	11	2
垃圾邮件数据集	4601	58	2
Wave	5000	22	2
Tae	151	3	3
Thy	215	5	3
光学字符识别数字	5620	63	2
混凝土	1030	9	2

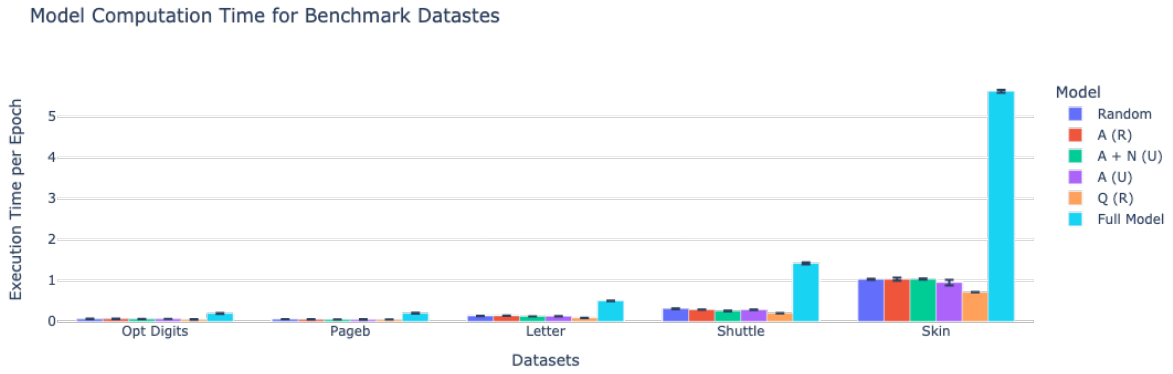
表 1 分类基准数据集：用于评估所提出模型分类检测能力的基准数据集描述。 *N* - 实例数量， *d* - 属性数量， *c* - 数据集中真实类别的数量。



(c)

6 致谢

作者谨此感谢布法罗大学计算研究中心（http://www.buffalo.edu/ccr.html）提供计算资源，以供开展本文所报告的研究。本文研究获得了美国国家科学基金会资助，基金编号为 NSF/OAC 1339765 和 NSF/DMS 1621853，谨此致谢。

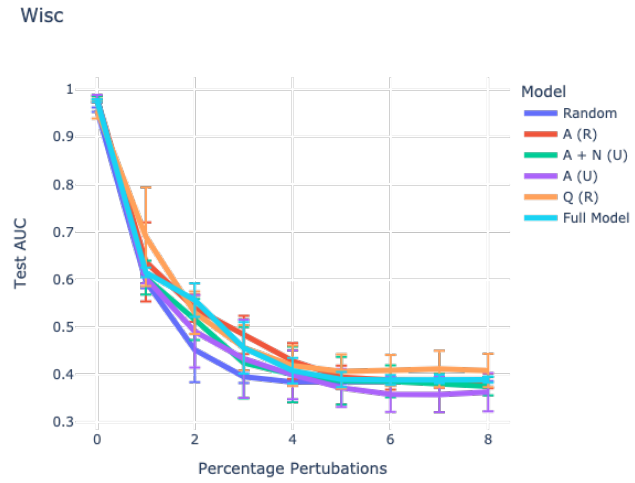


(d)

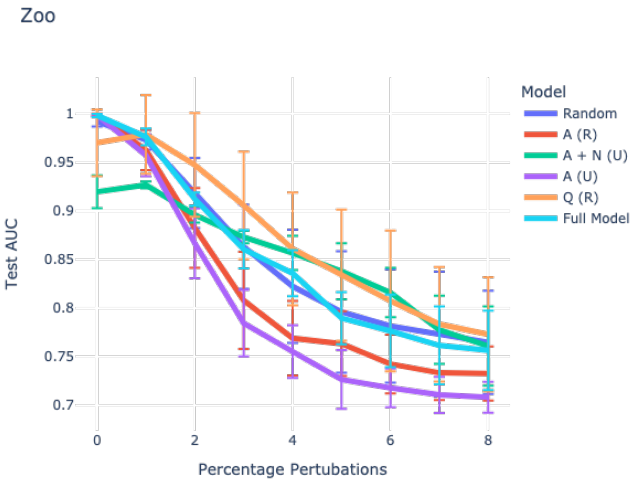
图4 不同数据集的计算时间：图中展示了不同 LIIT 训练的人工神经网络模型的计算时间，并与使用完整训练数据训练的人工神经网络（完整模型）进行了比较。

参考文献

[1] Dheeru, D. 和 E. Karra Taniskidou，2017：《style id='2'>UCI 机器学习知识库。网址：http://archive.ics.uci.edu/ml[2] Djolonga, J., J. Yung, M. Tschannen, R. Romijnders, L. Beyer, A. Kolesnikov, J. Puigcerver, M. Minderer, A. D'Amour, D. Moldovan 等，2021：卷积神经网络的鲁棒性与可迁移性。*IEEE/CVF* 计算机视觉与模式识别会议论文集，16458–16468。[3] Kavzoglu, T.，2009：使用精炼训练数据提高神经网络分类的准确性。《环境建模与软件》，**24**，第7期，850–858。[4] Schapire, R. E.，2003：机器学习中的提升法：综述。非线性估计与分类，149–171。[5] Shanker, M., M. Y. Hu 和 M. S. Hung，1996：数据标准化对神经网络训练的影响。*Omega*，**24**，第4期，385–397。[6] Soekhoe, D., P. Van Der Putten 和 A. Plaat，2016：深度神经网络迁移学习中数据集规模的影响。智能数据分析国际研讨会，Springer，50–60。[7] Wang, J., S. Li, Z. An, X. Jiang, W. Qian 和 S. Ji，2019：用于实现机器快速智能故障诊断的批归一化深度神经网络。*Neurocomputing*，**329**，53–65。[8] Zhong, Y., F. Fei, Y. Liu, B. Zhao, H. Jiao 和 L. Zhang，2017：SatCNN：使用敏捷卷积神经网络进行卫星图像数据集分类。《遥感快报》，**8**，第2期，136–145。


Wisc
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations

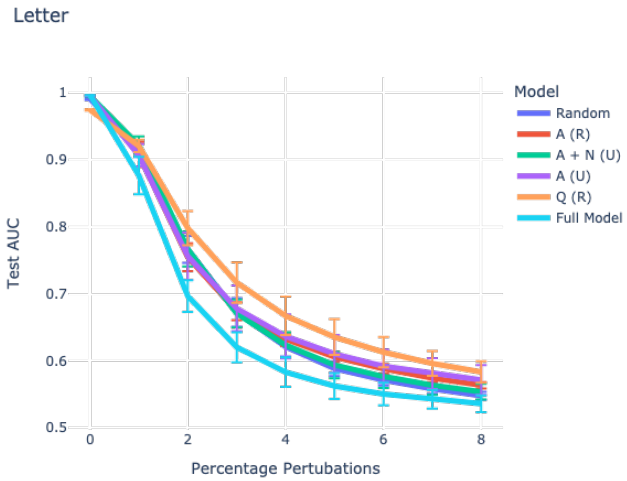
(a)


Zoo
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
Percentage Pertubations

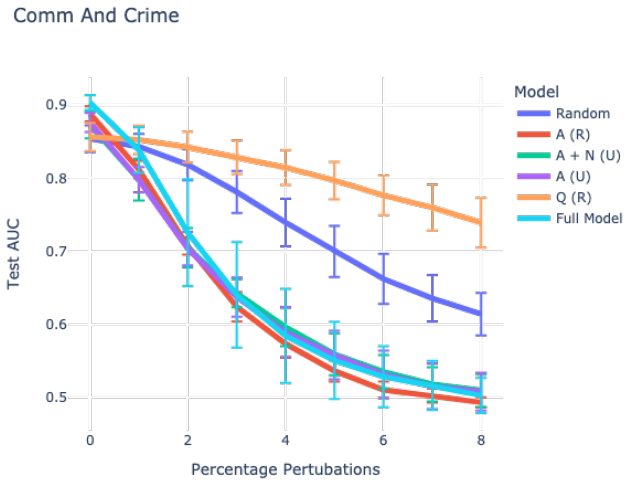
(b)

Data	Random	Anomaly (Repeated)	Anomaly + Normal (Unique)	Anomaly (Unique)	Quantile Samples (Repeated)	Full Model
Tae	0.6 (± 0.04)	0.54 (± 0.04)	0.72 (± 0.05)	0.6 (± 0.06)	0.63 (± 0.14)	0.63 (± 0.05)
Spambase	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.98 (± 0.01)	1.0 (± 0.0)
Comm And Crime	0.85 (± 0.02)	0.89 (± 0.01)	0.87 (± 0.02)	0.88 (± 0.01)	0.86 (± 0.02)	0.9 (± 0.01)
Wisc	0.96 (± 0.0)	0.98 (± 0.01)	0.98 (± 0.0)	0.98 (± 0.01)	0.96 (± 0.02)	0.98 (± 0.0)
Letter	0.99 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.97 (± 0.0)	1.0 (± 0.0)
Vowel	0.94 (± 0.01)	0.91 (± 0.01)	0.87 (± 0.02)	0.91 (± 0.01)	0.75 (± 0.05)	1.0 (± 0.0)
Pageb	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Thy	1.0 (± 0.0)	0.98 (± 0.02)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.01)	1.0 (± 0.0)
Zoo	0.99 (± 0.01)	1.0 (± 0.0)	0.92 (± 0.02)	1.0 (± 0.0)	0.97 (± 0.03)	1.0 (± 0.0)
Concrete	0.99 (± 0.0)	0.99 (± 0.01)	0.99 (± 0.0)	0.99 (± 0.01)	0.96 (± 0.02)	0.99 (± 0.0)
Wave	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Fault	0.98 (± 0.01)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.92 (± 0.01)	1.0 (± 0.0)
Shuttle	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Opt Digits	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.0)	1.0 (± 0.0)
Skin	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Imgseg	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.01)	1.0 (± 0.0)
Iono	0.95 (± 0.02)	0.97 (± 0.02)	0.93 (± 0.01)	0.97 (± 0.03)	0.83 (± 0.07)	0.99 (± 0.01)
Balance-Scale	0.97 (± 0.01)	1.0 (± 0.0)	0.99 (± 0.01)	0.99 (± 0.01)	0.88 (± 0.04)	1.0 (± 0.0)
Sonar	0.72 (± 0.08)	0.8 (± 0.05)	0.73 (± 0.03)	0.8 (± 0.05)	0.56 (± 0.12)	0.89 (± 0.02)
Ecoli	0.95 (± 0.01)	0.92 (± 0.03)	0.85 (± 0.03)	0.95 (± 0.02)	0.92 (± 0.02)	0.95 (± 0.0)

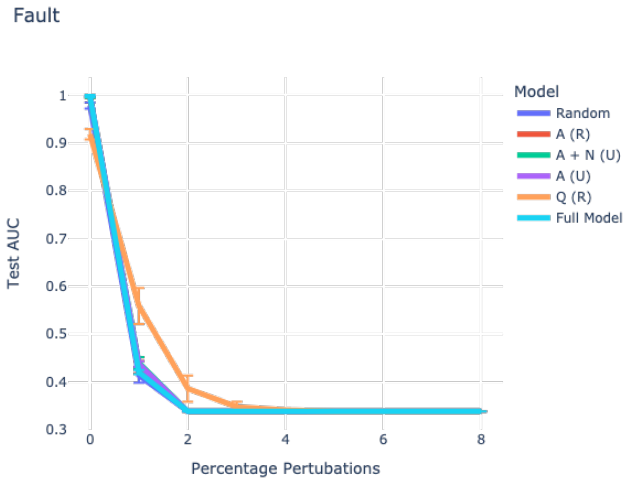
TABLE 2 Model Performance on Test Data: The mean AUC for each model on different datasets is shown here. The standard deviation in AUC on different re-runs is shown here.


Letter
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

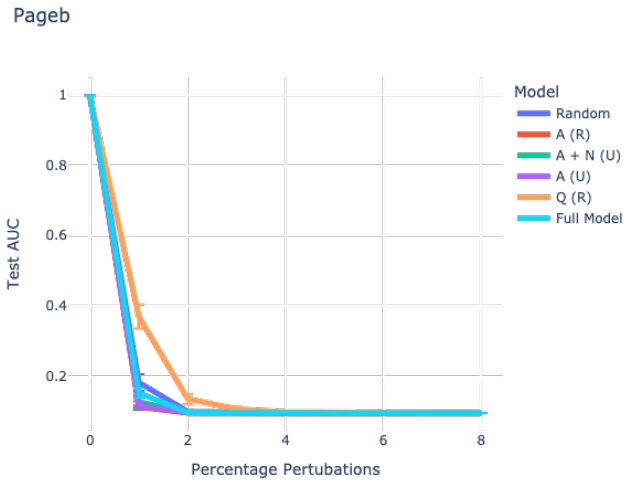
(c)


Comm And Crime
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

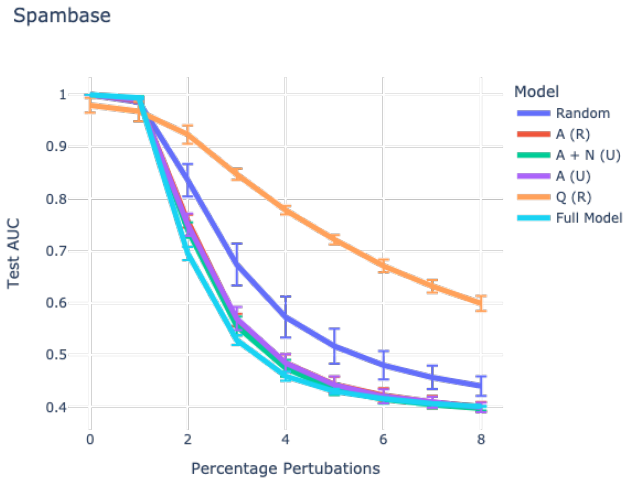
(d)


Fault
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

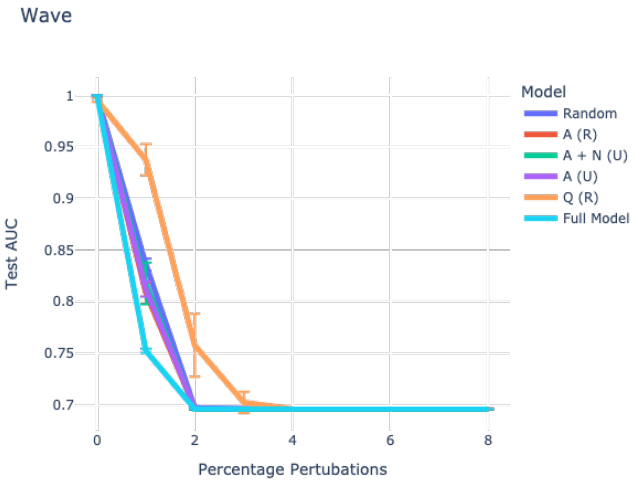
(e)


Pageb
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

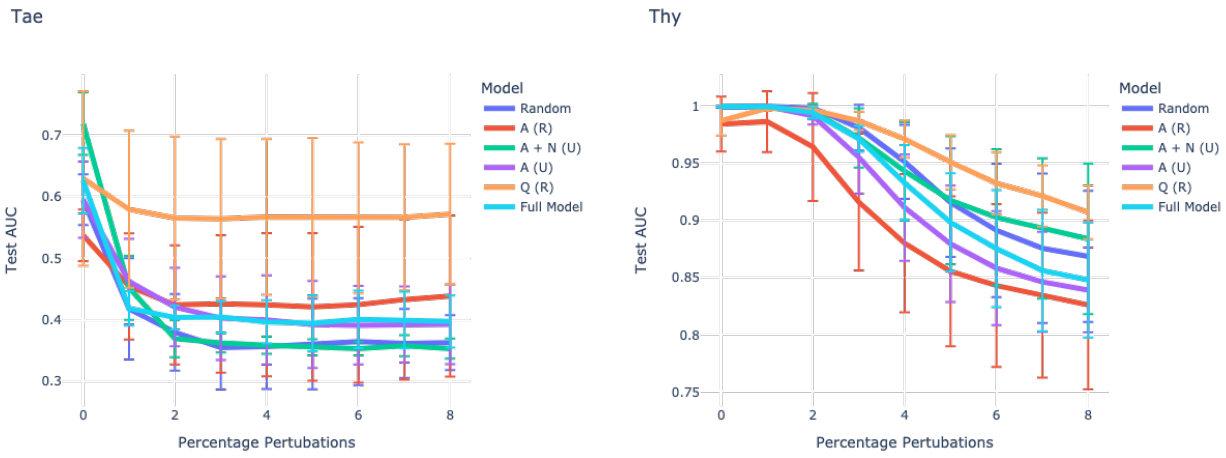
(f)


Spambase
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

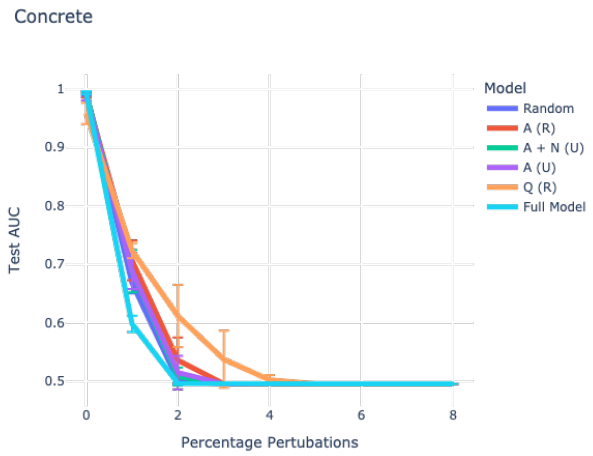
(g)


Wave
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

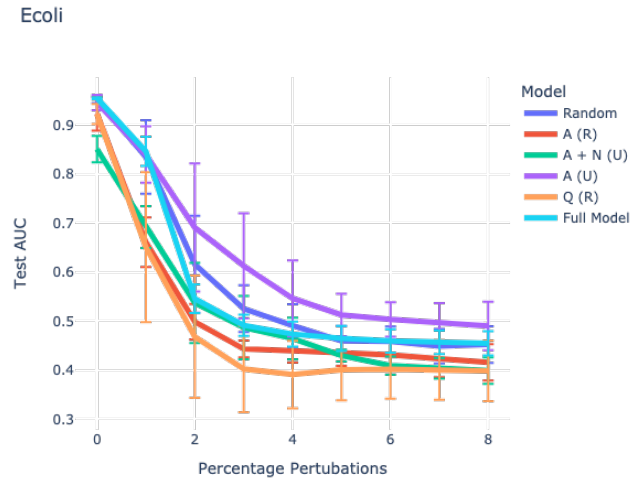
(h)


Tae
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Thy
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

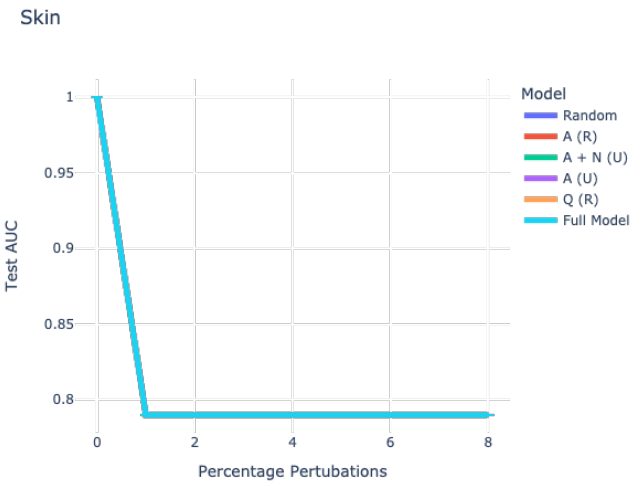
(i) (j)


Concrete
Test AUC
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

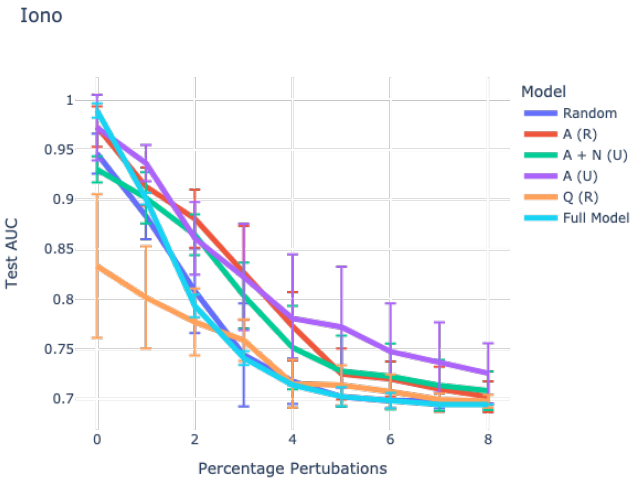
(k)


Ecoli
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations

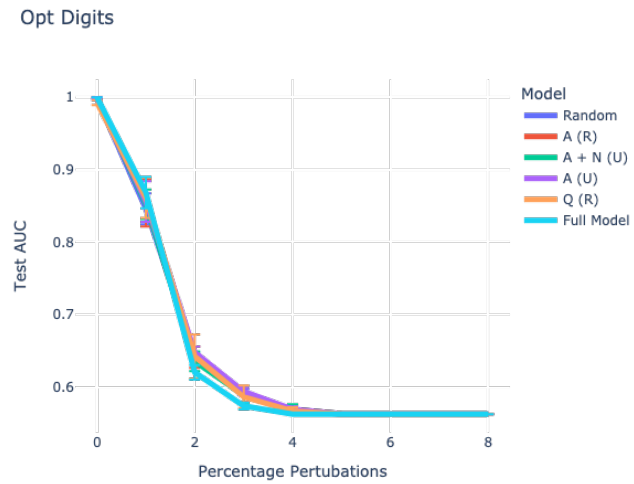
(l)


Skin
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0
2
4
6
8
Percentage Pertubations

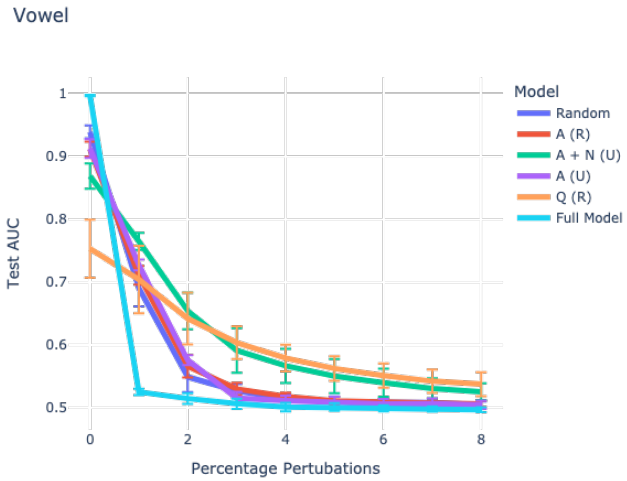
(m)


Iono
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
Percentage Pertubations

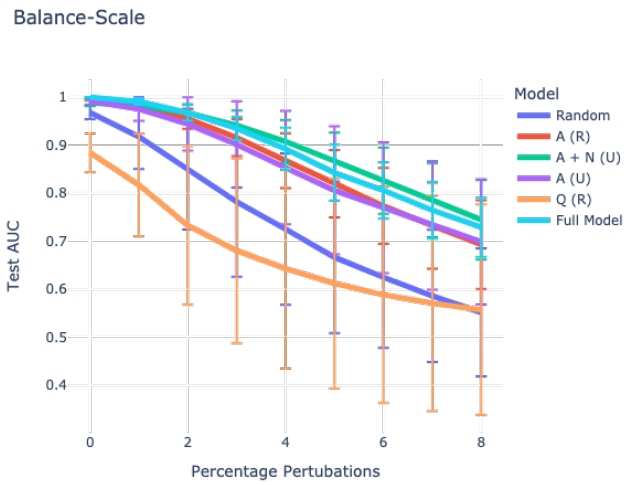
(n)


Opt Digits
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0
2
4
6
8
Percentage Pertubations

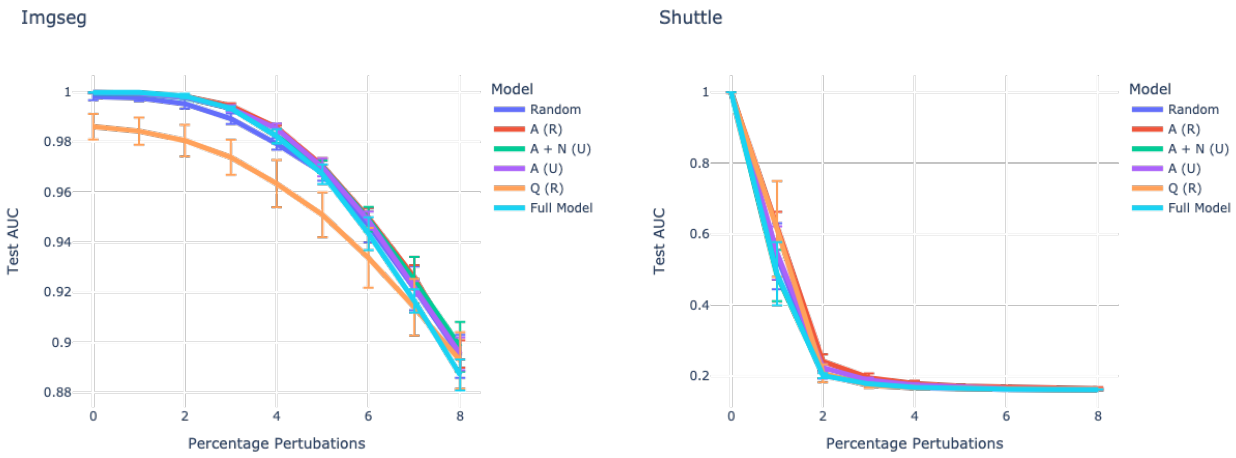
(o)


Vowel
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations

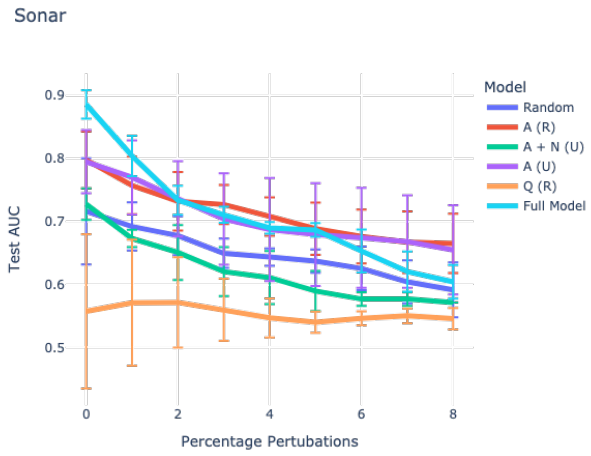
(p)


Balance-Scale
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0
2
4
6
8
Percentage Pertubations

(q)



(r) **(s)**

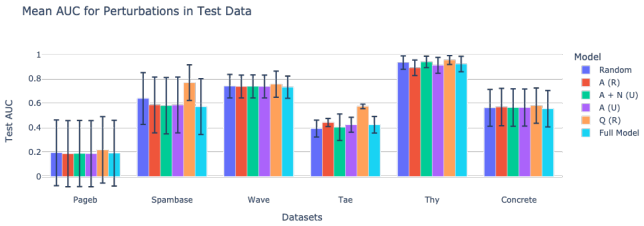


(t)

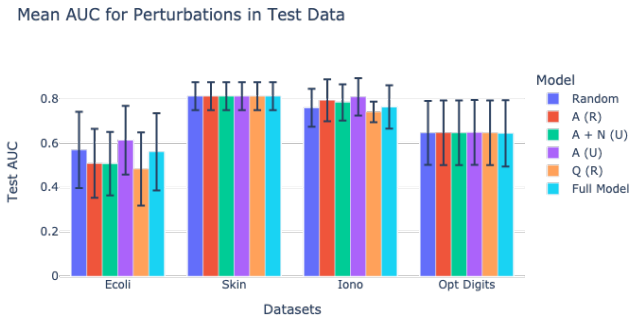
图 5 向数据中添加 % 扰动后 AUC 的变化：该图展示了随着向不同数据集添加 % 扰动，AUC 的变化。误差条表示 5 次重复实验中 AUC 值的标准差



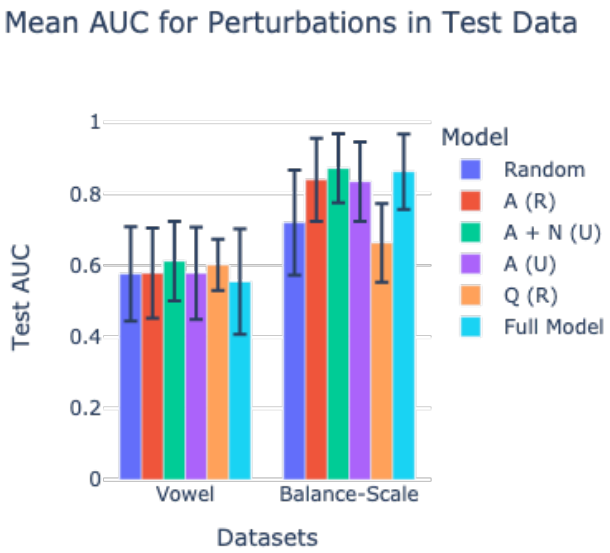
(a) 分位数采样 LIIT 训练模型性能最佳的数据集



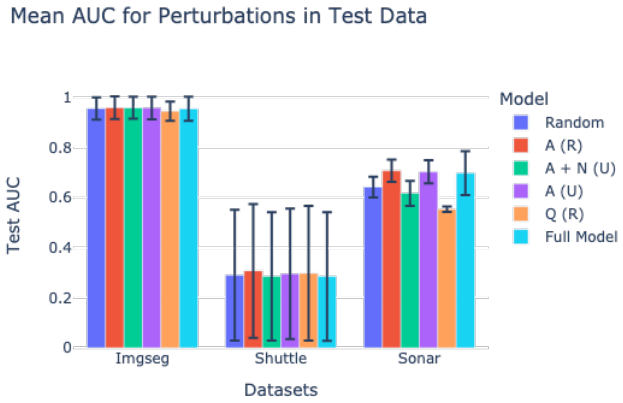
(b) 分位数采样 LIIT 训练模型性能最佳的数据集



(c) 异常（唯一）LIIT 训练模型性能最佳的数据集



(d) 异常 + 正常（重复）LIIT 训练模型性能最佳的数据集



(e) 异常（重复）LIIT 训练模型性能最佳的数据集

图 6 向数据中添加 % 扰动后 AUC 的总变化：该图展示了随着向不同数据集增加 % 扰动，AUC 均值的变化。误差条表示各扰动下 AUC 值的标准差。